

LAB4

1. Байгуулагч функц гэж юу вэ?

Нэр нь классын нэртэй адил байх гишүүн функцыг класс дотор нэмж тодорхойлон объектод ой бэлдэх гарааны утга оноох зэрэг үйлдэл хийлгэж болдог. Энэ функцыг байгуулагч функц гэх ба ямар ч утга буцаахгүй, класс дотор үүсгэж өгөөгүй бол компайлер өөрөө үүсгэдэг (далдуур). Байгуулагч функц объект үүсэх үед хамгийн эхэнд автоматаар дуудагдаж ажилладаг.

2. Устгагч функц гэж юу вэ?

Объект устгагч функцгүйгээр устаж чадахгүй. Байгуулагч функцаар бэлдсэн ойг чөлөөлөхөд устгагч функц хэрэглэгддэг ба устгагч функцын нэр нь `~classname` байна. Класс олон байгуулагч функцтай байж болох ч ганцхан устгагч функцтай байна.

3. Функц дахин тодорхойлох гэж юу вэ? Жишээгээр тайлбарал.

Функц дахин тодорхойлох гэдэг нь нэг нэртэй хэдий ч авах `argument`-н тоо төрөл дараалал нь ялгаатай байх олон функц тодорхойлохыг хэлнэ. Өөр нэг нэршил нь **polymorphism**. Энэ нь ерөнхийдөө бичдэс нь ялгаатай байдлаар тодорхойлогдсон адил нэртэй функцууд юм.

Жишээ нь:

```
void printSum(int a,int b){
    cout<<a+b;
}

void printSum(int a,int b,int c){
    cout<<a+b+c;
}
```

Энэ хоёр адил нэртэй ч нэг нь 2 тоо нэмж хэвлэх бол нөгөө нь 3 тоо нэмж хэвлэнэ. Бичдэс нь өөр юмуудаа, C дээр бол print function зарлаад хэвлэх үйлдлээ хялбар болгож болно.

4. Динамик санах ой (new, delete оператор ашигласан) болон байгуулагч, устагч функцийг хэрхэн хамтад нь ашиглах вэ? Хэрэгжүүлэлт

```
class person{
private:
    string name;
    int age;
    int* childNumber;
    char childrenNames;
public:
    person(){
        childNumber = new int;
        char childrenNames = new char[100];
    }
    ~person(){
        delete childNumber;
        delete childrenNames;
    }
}
```

Динамик санах ойг (new, delete) байгуулагч функц дээр нөөцлөн авч, түүнийгээ заавал устгагч функц дээр чөлөөлөх шаардлагатай хэрэв чөлөөлөхгүй бол санах ойн цоорхой үүсэх эрсдэлтэй. Дээр кодоор хэрэгжүүлэв.

1. Lab03-д тодорхойлсон класст анхдагч болон параметртэй байгуулагч нэмж тодорхойлох ба тодорхойлохдоо байгуулагч функцийн үүргийг сайтар хэрэгжүүлсэн байх шаардлагатай. ,4,5

```

#include <iostream>
using namespace std;
class worker{
    int number;
    float hours;
    string name, rank;
    float directorSalary(float salary){
        return salary+=500000;
    }
public:
    //constructors
    worker(){
        number=0;
        name="";
        rank="";
        hours=0;
    }
    worker(int outNumber, const string outName,const string outRank,float outHours)
    {
        number=outNumber;
        name=outName;
        rank=outRank;
        hours=outHours;
    }
    worker(worker& employee){
        number=employee.number;
        name=employee.name;
        rank=employee.rank;
        hours=employee.hours;
    }
    void setEmployee(int num,string oName,string oRank,float oHours){
        number=num;
        name=oName;
        rank=oRank;
        hours=oHours;
    }
    void getEmployee(){
        cout<<"\nname : "<<name;
        cout<<"\nrank : "<<rank;
        cout<<"\nhours : "<<hours;
        cout<<"\nnumber : "<<number;
    }
    float salary(){
        float salary;
        salary=12000.0*hours;
        if(rank=="zahiral")salary+=directorSalary(salary);
        return salary;
    }
    bool timeIncreaser(float addedTime){
        if(addedTime>0 && addedTime<24){
            hours+=addedTime;
            return 1;
        }
    }

```

```

        }else{
            return 0;
        }
    }
};

int main() {
    worker employees[3];
    for(int i=0;i<3;i++){
        int number;
        float hours;
        string name, rank;
        cin>>number>>name>>rank>>hours;
        employees[i].setEmployee(number, name, rank, hours);
    }
    for(int i=0;i<3-1;i++){
        for(int j=0;j<3;j++){
            if(employees[j].salary<employees[i]){
                worker temp(employees[j]);
                employees[i] = employees[j];
                employees[j] = temp;
            }
        }
    }
    for(int i=0;i<3;i++){
        employees[i].getEmployee();
    }
    return 0;
}

```

2. Lab03-д тодорхойлсон класст устгагч функц тодорхойлж устгагч функц хэзээ дуудагдаж байгааг туршилтад үндэслэн хариул.

Устгагч функц нь програм ажиллаж дуусахад дуудагдаж байна. Мөн функцын локал хувьсагч байх үедээ тухайн функцын үйлчлэх хязгаар {} дуусахад дуудагдаж байна.